

```

/**
 * @author >> Justin Windle wrote the underlaying motion detection program
 * @link >> soulwire.co.uk
 * @modified by >> Gareth Dorset, Jun Cui, Trevor Farquharson
 */

/*
Ant Project. Otago Polytechnic, New Zealand. 2009 3rd Year B.I.T. project for
Jun Cui, Gareth Dorset and Trevor Farquharson
-:~*._.*~:-

Client: Otago Museum

Developers: Jun Cui - 3rd year student Otago Polytechnic
            Gareth Dorset - 3rd year student Otago Polytechnic
            Trevor Farquharson - 3rd year student Otago Polytechnic

Mentors: Patricia Haden - Otago Polytechnic
         Hamish Smith - Otago Polytechnic
         Sam Mann - Otago Polytechnic

*/

/* The code contained below is modified from the original motion detection
code by Justin Windle, full credit for making the motion tracker work goes to him.

We had full permission from Justin Windle to use and modify this code.*/
package
{
    import com.soulwire.geom.ColourMatrix;
    import com.soulwire.media.MotionTracker;
    import flash.display.*;
    import flash.events.Event;
    import flash.filters.ColorMatrixFilter;
    import flash.media.Camera;
    import flash.media.Video;

    public class TPerson extends Sprite
    {
        // easing for the motion detector target
        private const _ease: Number = 5;

        private var _motionTracker: MotionTracker;

        private var _noMotionCounter: Number = 0;
        private var _noMotion: Boolean = true;

        private var _target: Shape;
        private var _bounds: Shape;
        private var _vidOutput: Bitmap;
        private var _vidSource: Bitmap;
        private var _video: BitmapData;
        private var _matrix: ColourMatrix;

        private var screenWidth: Number;
        private var screenHeight: Number;

        // variables to pass into ant class
        private var _personX: Number;
        private var _personY: Number;

        public function TPerson(frames: Number, sW: Number, sH: Number)
        {
            screenWidth = sW;
            screenHeight = sH;
            var camW: Number = screenWidth;
            var camH: Number = screenHeight;

            // create camera
            var cam: Camera = Camera.getCamera();
            cam.setMode(camW, camH, frames);

            // create video
            var vid: Video = new Video(camW, camH);
            vid.attachCamera(cam);

            // create motiontracker
            _motionTracker = new MotionTracker(vid);

            // flip input
            _motionTracker.flipInput = false;

            // motion visualisation stuff
            // make the colour matrix
            _matrix = new ColourMatrix();
            _matrix.brightness = _motionTracker.brightness;
            _matrix.contrast = _motionTracker.contrast;

            //display camera input with filters the motion tracker is using
            _video = new BitmapData(camW, camH, false, 0);
            _vidSource = new Bitmap(_video);
            _vidSource.scaleX = -1;
            _vidSource.x = camW;
            _vidSource.filters = [new ColorMatrixFilter(_matrix.getMatrix())];

```

```

        //processed output display stuff. --> not displaying to screen
        _vidOutput = new Bitmap(_motionTracker.trackingImage);
        // _vidOutput.scaleX = -1;
        _vidOutput.x = 0;

        // target displaying movement
        _target = new Shape();
        _target.graphics.beginFill(0xFFFFF00);
        _target.graphics.lineStyle(0, 0xFFFFF00);
        _target.graphics.drawCircle(0,0,5);
        _target.graphics.endFill();
        // addChild(_target); // display the target temporarily.
        // bounding box displaying movement
        _bounds = new Shape();
        _bounds.x = _vidOutput.x;
        _bounds.y = _vidOutput.y;

        // add event listener for tracking
        addEventListener(Event.ENTER_FRAME, track);
    } // End Constructor

    private function track(e:Event): void
    {
        // tell the tracker to update
        _motionTracker.track();

        if(_noMotion) {
            target.x = -500;
        } else {
            // move circle with easing
            _target.x += (((_motionTracker.x) + _bounds.x) - _target.x) / _ease;
            _target.y += (((_motionTracker.y) + _bounds.y) - _target.y) / _ease;
        }

        // draw to screen
        _video.draw(_motionTracker.input);

        // set my personX and personY variable
        _personX = _target.x;
        _personY = _target.y;

        if(_noMotionCounter > 120)
        {
            _noMotion = true;
            _noMotionCounter = 0;
        }

        // if movement detected, continue, otherwise, leave method
        if(!_motionTracker.hasMovement)
        {
            _noMotionCounter++;
            return;
        }
        _noMotionCounter = 0;
        _noMotion = false;

        _bounds.graphics.clear();
        _bounds.graphics.lineStyle( 0, 0xFF0000 );
        _bounds.graphics.drawRect( (_motionTracker.motionArea.x),
                                   _motionTracker.motionArea.y,
                                   _motionTracker.motionArea.width,
                                   _motionTracker.motionArea.height
                                   );

    } // End Track Method

    public function GetXLoc(): Number
    {
        return _personX;
    } // End GetPersonX

    public function GetYLoc(): Number
    {
        return _personY;
    } // End GetPersonY
} // End TPerson Class
} // End Package

```